Trabajo de Fin de Grado

Grado en Ingeniería Informática

Curso 2018/2019

Facultad de informática

UNIVERSIDAD
COMPLUTENSE
MADRID

Desarrollo y Securización de una Herramienta de Autodefensa Digital

Kapi Solutions

Autor: **Javier Martín Villarreal**
Tutor: **Marcos Sánchez-Elez Martín**

Lista de Contenidos

1 Introducción **10**
1.1 Objetivos . . . 10
1.1.1 Mejorar la accesibilidad a la seguridad online . . . 10
1.1.2 Contribuciones como usuario . . . 10
1.1.3 Contribuciones como desarrollador . . . 11
1.1.4 Contribuciones a la mejora de seguridad de la aplicación 11
1.2 Plan de trabajo . . . 12
1.3 Metodología de trabajo . . . 12
1.3.1 Contribución en el front-end . . . 13

2 Arquitectura de la aplicación **14**
2.1 Decisión de herramientas . . . 14
2.2 Virtualización de máquinas . . . 15
2.2.1 Herramientas de virtualización . . . 15
2.2.2 Análisis de las herramientas . . . 15
2.2.3 Conclusiones . . . 16
2.3 Integración de Kapi en Docker . . . 17
2.3.1 Infraestructura de desarrollo . . . 20

3 Implementación de la aplicación web **22**
3.1 Árbol de directorios . . . 22
3.2 Diseño de la base de datos . . . 22
3.3 Contribuciones sobre el backend . . . 22
3.3.1 Conexión y desconexión de la base de datos . . . 23
3.3.2 Búsqueda en la aplicación . . . 23
3.3.3 Mostrar creación de un artículo . . . 25
3.3.4 Mostrar artículo publicado . . . 26
3.3.5 Crear un artículo . . . 26
3.3.6 Generar exámenes . . . 27
3.3.7 Guardar examen y nota . . . 27
3.3.8 Información en el perfil de usuario . . . 28
3.3.9 Hasheo de la contraseña . . . 28
3.4 Población masiva de datos . . . 29

4 Securización de la aplicación **29**
4.1 Análisis de vulnerabilidades . . . 29
4.2 Vulnerabilidades resueltas . . . 30
4.2.1 Falla por inyección de SQL . . . 31
4.2.2 Exploración de directorios . . . 32
4.2.3 X-Frame-Options Header Not Set . . . 33
4.2.4 Cross-Domain JavaScript Source File Inclusion . . . 33

5 Resultados y discusión **34**
5.1 Uso de la aplicación web 34
5.2 Uso como desarrollador 36

6 Conslusiones **37**
6.1 Mejora de la accesibilidad a la seguridad digital 37
6.2 Ofrecer contribuciones como usuario 37
6.3 Ofrecer colaboración como desarrollador 37
6.4 Garantizar mejoras de seguridad 38

7 Bibliografía **39**
7.1 Herramientas 39
7.2 Documentación e investigación 39

8 English version **40**
8.1 Introduction 40
8.1.1 Objectives 40
8.1.2 Work plan 42
8.1.3 Working Methodology 42
8.2 Conslusions 43
8.2.1 Improved the digital security accesibility 44
8.2.2 Offer user contributions 44
8.2.3 Offer development contributions 44
8.2.4 Guarantee security improvements 45

9 Apéndice **46**
9.1 Capturas de pantalla de la aplicación 46

Lista de Tablas

1 Distribucion de software entre máquinas 15
2 Puertos de la máquina Apache + PHP 19
3 Puertos de la máquina MySQL 19
4 Puertos de la máquina PHPMyAdmin 19

Lista de Figuras

1 Comparativa de recursos VM vs Docker 17
2 Esquema de la virtualización de los contenedores 20
3 Consumo de recursos de los contenedores 21
4 Árbol de directorios 23
5 Esquema relacional de las entidades de la base de datos . . . 24
6 Captura de la alerta de inyección SQL 31
7 Captura de la alerta de Exploración de directorios 33
8 Signin 46
9 Login 46
10 Artículos 47
11 Modificaciones 47
12 Creaciones 47
13 Crear artículo 48
14 Modificación 48
15 Perfil 49
16 Exámenes 49
17 Examen 49
18 Alertas detectadas con OWASP ZAP 50

Lista de Códigos

1 docker-compose.yml 20
2 backend/conexion.php 23
3 backend/desconexion.php 23
4 backend/busqueda.php 24
5 backend/busquedaarticulos.php 24
6 backend/busquedacreaciones.php 24
7 backend/busquedamodificaciones.php 25
8 bakend/busquedausuarios.php 25
9 backend/busquedapreguntas.php 25
10 backend/creacion.php 26
11 backend/articulo.php 26
12 backend/creararticulo.php 26
13 backend/subeImgArticulo.php 26

14 backend/generaexamen.php 27
15 backend/guardarExamen.php 27
16 backend/usuario.php 28
17 backend/creacionesusuario.php 28
18 backend/modificacionesusuario.php 28
19 backend/publicacionesusuario.php 28
20 backend/signin.php 29
21 backend/login.php 29
22 SQL inyetion vulnerable 31
23 SQL inyection resuelto 31
24 SQL inyetion vulnerable 32
25 SQL inyection resuelto 32
26 SQL inyetion vulnerable 32
27 SQL inyection resuelto 32
28 SQL inyetion vulnerable 32
29 SQL inyection resuelto 32
30 X-Frame-Option Header Not Set resuelto 33
31 Cross-Domain JavaScript Source File Inclusion 34
32 Cross-Domain JavaScript Source File Inclusion resuelto . . . 34

Agradecimientos

Quiero agradecer a mi **familia** y amigos todo su apoyo en mi trayectoria universitaria.

Tanto a mis padres como a mis hermanas y aquellas personas que me han apoyado, ayudado y animado en los momentos difíciles y que han estado también en los fáciles, y a aquellos que no han estado pero les habría encantado poder estar más cerca.

Quiero agradecer también a mis compañeros y profesores de universidad todo lo que me han enseñado tanto con su ayuda y su esfuerzo, como aquellos que no han colaborado y me han obligado a esforzarme aún más para poder llegar a este momento.

Por esto doy las gracias de todo lo aprendido (bueno y malo) ya que en estos años he crecido tanto como persona, como en conocimientos.

Resumen

En esta sección resumiré las partes que he desarrollado dentro de este proyecto y la finalidad del mismo.

Mi trabajo puede distinguirse en tres fases:

- **Implementación de una herramienta web orientada al aprendizaje y divulgación de la seguridad informática.**

  Kapi Solutions es una aplicación web en la que el usuario podrá aprender y ayudar en el ámbito de la seguridad digital en función de su sector profesional. Esto significa que:

  - El usuario podrá aprender de la información proporcionada por sus colegas en la web, independientemente de su nivel de conocimiento en el campo de la seguridad digital ya que proporciona un método de asignación de nivel.
  - El usuario podrá aprender y apoyar a la comunidad con aportaciones de código sobre la herramienta.
  - El usuario podrá ayudar a sus colegas aportando información con artículos y preguntas.
  - El usuario podrá apoyar a la comunidad aportando votos positivos o negativos a los artículos que están en fase de revisión, así como sugiriendo modificaciones de aquellos que ya existan.

- **Integración de la herramienta para portabilizarla y facilitar su desarrollo y despliegues.**

  El desarrollo compartido de una herramienta es complicado, para poder ofrecernos a una comunidad de desarrollo y poder tener la aplicación web portable de la manera más cómoda posible, tendré que decidir qué subsistemas de virtualización o alternativas existen.

- **Analizar la herramienta en busca de vulnerabilidades para securizarla.**

  Una vez terminada la aplicación y debidamente desplegada, analizaré las vulnerabilidades que ofrece y resolveré aquellas necesarias.

Abstract

In this section, I will sume the parts I have developed on this proyect and it's goals.

My work could be splitted into three parts:

- **Development of a web based tool focused on learning and divulge about digital security.**

  Kapi Solutions is a web application where the user could learn and help about digital security oriented to his/her profession skills. That means:

  - The user could learn with the information provided by his/her collegues on the web, regardless of his/her knowledge level about the digital security field because we provided a method to reach your real level.
  - The user could learn and support the community with code contributions to the tool.
  - The user could help his/her collegues supporting with new articles and questions.
  - The user could support the community voting the articles on the review stage, such as uploading new modifications on the existing ones.

- **Portable tool integration to facilitate development and deployment.**

  Sharing a development over one tool, is hard, if we want to offer us to a development community and port the tool in the easiest way, I will choose over virtualizing tools and other alternatives to get this way.

- **Analize the tool, looking for security vulnerabilities to solve them and make the tool more secure.**

  Once the tool is finished and deployed, I will analize it to find security vulneravilities, I will study them and solve the most of them.

Palabras clave

A continuación detallo las palabras clave separadas por comas:

Kapi, solutions, artículo, artículos, creación, modificación, comunidad, desarrollo, profesional, rol, nivel, examen, docker, aplicación, web, back-end, base, datos, servicio, máquina, puerto, configuración, seguridad, digital, cliente, servidor, vulnerabilidad.

Keywords

In the following, there is a detail of the keywords separated by comas:

Kapi, solutions, article, articles, creation, modification, community, develop, professional, role, level, exam, docker, application, web, back-end, base, data, service, machine, port, configuration, security, digital, client, server, vulnerability.

1 Introducción

En esta sección veremos tanto la motivación del proyecto como los objetivos a alcanzar y las herramientas similares que existen.

1.1 Objetivos

Para resolver los problemas que hemos mencionado, debemos plantear los objetivos que se describen a continuación.

1.1.1 Mejorar la accesibilidad a la seguridad online

Como se ha mencionado antes, gracias a internet podemos comunicarnos e informarnos de gran cantidad de temas, sin embargo estos son demasiado técnicos así que **Kapi**, debe ser una herramienta en la que la información está escrita por y para profesionales de un sector específico. Empezaremos enfocándonos en tres sectores (**abogados**, **periodistas** e **informáticos**). De esta manera queremos romper con el problema de que un abogado o periodista, en este caso, no sean capaces de comprender lo que necesitan para resolver su duda porque los artículos que encuentren no estén redactados con la información que realmente necesita para su profesión si no que son demasiado densos y técnicos o por el contrario, comentarios cortos que son difíciles de comprender.

Para mejorarlo, queremos crear un entorno en el que cada artículo esté redactado para un perfil profesional concreto, éste será redactado únicamente por usuarios que pertenecen a ese perfil profesional y antes de ser validado, deberá pasar por una fase de validación por parte de la comunidad, así será sometido a revisiones y modificaciones hasta coger su forma final completa que se publicará.

En cualquier caso, tanto un artículo en fase de revisión y uno publicado, tendrán la posibilidad de ser modificados por otro usuario con el mismo rol.

1.1.2 Contribuciones como usuario

La herramienta ofrecerá las vías necesarias para que cualquier usuario pueda obtener información del campo que quiera, pero solo pueda crear artículos del rol profesional al que el mismo usuario pertenece. Igualmente proporcionaremos un sistema de niveles para poder clasificar a los usuarios a partir de exámenes de manera que estos, sólo puedan crear/modificar contenidos de como máximo su nivel.

Así los usuarios no podrán modificar artículos de otros usuarios con nivel superior y tendrán la oportunidad de adquirir ese nivel realizando pruebas

de examen.

Las modificaciones y creaciones, antes de publicarse, se someten a una fase de votaciones, donde los usuarios deciden qué artículos y modificaciones de artículos se publican.

Por tanto, las contribuciones como usuario serán:

- Crear artículos de como máximo su nivel.
- Modificar artículos de su rol, de nivel inferior o igual al del usuario.
- Votaciones para decidir qué artículos se publican y cuáles no.

1.1.3 Contribuciones como desarrollador

Kapi, también ofrece la facilidad a los desarrolladores internos del proyecto de trabajar cómodamente en equipo, así como la oportunidad a cualquiera que quiera colaborar en una comunidad de software libre, una vía fácil de comprender para descargar el código fuente, lanzarlo en un entorno realista para hacer las pruebas pertinentes antes de aceptar el código del colaborador en el proyecto principal.

Para esto la herramienta dispondrá de acceso público a un repositorio basado en un controlador de versiones y una manera de virtualizar el servicio de modo sencillo y comprensible.
En resumen, el desarrollador dispondrá de:

- Repositorio público con control de versiones.
- Método de virtualización del proyecto para entorno de pruebas.

Con una virtualización cómoda y ligera, conseguiré que el desarrollo sea más homogéneo y portable. De esta manera el desarrollo será más cómodo y estable.

Además buscaré la forma más ligera y económica para facilitar los despliegues reales de la plataforma.

1.1.4 Contribuciones a la mejora de seguridad de la aplicación

Analizaré la herramienta en busca de vulnerabilidades de seguridad con herramientas de auditoría de seguridad web, en busca de vulnerabilidades para así solucionarlas y garantizar la mayor protección a los usuarios y un método de resolución de vulnerabilidades a la comunidad de desarrollo.

1.2 Plan de trabajo

El problema principal a abatir por este proyecto, es mejorar la accesibilidad a la información sobre seguridad online que encontramos en internet.

Internet dispone de gran cantidad de información muy buena y muy útil, pero en ocasiones nos encontramos que esa información viene explicada de una manera muy técnica y a veces compleja de entender incluso para personas habituadas al uso de las nuevas tecnologías.

Para esto, **Kapi** es una web en la que diferenciamos a los usuarios por roles profesionales, así cada usuario encontrará información redactada por gente de su gremio, lo que aporta un lenguaje más coloquial sobre los tecnicismos, explicaciones más personalizadas (el conocimiento en los gremios suele ser similar, de este modo, el que redacta el artículo, explicará los tecnicismos más complejos a los lectores de su gremio de manera que sea más comprensible).

Kapi es una herramienta orientada en lo posible a la comunidad libre. Para conseguir esto, desarrollaremos toda la herramienta con software libre, con imágenes y contenido con licencia libre y todo será público en un repositorio **git**, en nuestro caso alojado en GitHub.

Este repositorio debe estar preparado para trabajar cómodamente con él y que el colaborador sea capaz de probar sus cambios en su sistema de manera realista.

Quedando la planificación siguiente:

1. Mejorar la accesibilidad a la información sobre seguridad online.
2. Diferenciar los usuarios por roles profesionales, aportando una comunicación más cómoda y adaptada al gremio.
3. Implementar una metodología de desarrollo para la comunidad de software libre.
4. Implementar un despliegue portable y uniforme para desarrollar sobre una arquitectura real.

1.3 Metodología de trabajo

Esta aplicación web ha sido desarrollada por dos alumnos de la Facultad de Informática de la Universidad Complutense de Madrid, **Álvaro Asenjo Torrico** y **Javier Martín Villarreal**.

Por motivos académicos, nos planteamos desarrollar este proyecto con vistas a presentarlo en años consecutivos. Desde el principio trabajamos juntos separando las tareas de forma clara pues mi compañero presentaría un año antes que yo y queríamos que las tareas quedasen bien diferenciadas por este motivo.

Es por esto que mi compañero se encargaría de desarrollar el **front-end**.

Por otro lado yo me encargaría de las tareas de **implementación de la arquitectura** del proyecto, así como la **estructura del árbol de directorios** que implementaremos para el desarrollo, el **back-end** de la aplicación (aquellas zonas de código que se ejecutan en el servidor para devolver los datos al cliente), la definición de la **base de datos**, la **población de datos** de prueba sobre la base de datos (para tener una idea del resultado de la web con contenido), la **virtualización** de esta arquitectura para su portabilidad y el estudio de la **seguridad** tanto del sistema virtualizado, como de la aplicación.

Solo haré alguna mención sobre el trabajo desarrollado por mi compañero y trataré de detallar únicamente las tareas que he desarrollado yo sobre esta herramienta.

Entre los dos decidimos el flujo que íbamos a desarrollar para poder acceder a los artículos de la web, poder votar los artículos propuestos por la comunidad, poder realizar exámenes en la plataforma para ascender por los niveles y la gestión de los datos de usuario, tanto modificación como visualización de los mismos.

A continuación explico en detalle el desarrollo de las tareas desarrolladas por mi compañero.

1.3.1 Contribución en el front-end

Mi compañero, se encargó de desarrollar el frontal de la aplicación, todas aquellas partes que se ejecutarán en el cliente.

Para ello, decidió implementarlo con PHP, HTML, JavaScript, CSS y Bootstrap para estilar la página.

Desarrolló todas las vistas de la aplicación y decidimos las variables y métodos que utilizaríamos para la **comunicación** entre el front-end y el back-end, comunicando de manera uniforme el cliente con el servidor.

2 Arquitectura de la aplicación

En este primer capítulo, decidiremos las bases de desarrollo de la plataforma y estudiaremos los motivos de elección de cada decisión.

2.1 Decisión de herramientas

Para montar el servidor, facilitar el trabajo de la comunidad y mantener una uniformidad de desarrollo, tenemos que definir una infraestructura sólida y muy clara. Ya que kapi se ha desarrollado con varias herramientas y servicios como **PHP**, **apache**, una base de datos **mysql** y lenguajes web **html**, **css**, **js** y **bootstrap**.

Para soportar estas tecnologías diferenciamos entre las cargas que soportarán los servidores y las que recaerán sobre los clientes; así separamos el servicio apache junto con PHP y la base de datos mysql como las cargas que recaerán sobre los servidores y los clientes soportarán la carga html, css y js.

Nos centraremos en las herramientas que trataremos en el servidor; estas también podemos separarlas en dos servicios diferenciados, apache que sirve los ficheros necesarios al cliente y mysql que sirve la base de datos. Estos servicios, separados en dos máquinas aportarán un extra de seguridad a priori.

A partir de este análisis, se plantean varias posibilidades para escalar el desarrollo de la plataforma. Analizaremos las posibilidades para tomar la más adecuada.

Para el desarrollo de la plataforma, el desarrollador debe de ser capaz de montar el entorno de la manera más sencilla y homogénea entre todos ellos. Para ello deberemos elegir un sistema operativo, versiones de servicios concretas y otras decisiones que iremos tratando.

Una vez elegido el stack de herramientas que vamos a utilizar, decidimos separarlas entre las máquinas que participarán en la actividad.

Como queremos montar estos servidores en un entorno aislado de desarrollo, encontramos dos ideas básicas: montar el sistema entero con dos servidores y una máquina que haga la función de cliente, o virtualizar las máquinas para que el desarrollo se haga sobre una única máquina.

Tabla 1: Distribucion de software entre máquinas

Servidor0	**Servidor1**	**Servidor3**	**Cliente**
Apache PHP	mySQL	PHPMyAdmin	HTML JavaScript CSS

Por motivos de infraestructura hardware, requisitos de potencia y complejidad de colaboración en la plataforma, es claramente más viable la opción de la virtualización de estas máquinas.

2.2 Virtualización de máquinas

En esta sección vamos a tratar las diferentes herramientas para virtualizar máquinas y decidiremos cuál será la elegida para el desarrollo del proyecto y por qué.

2.2.1 Herramientas de virtualización

A la hora de virtualizar máquinas nos encontramos con los diferentes hipervisores de sistemas que existen y son usados hoy día para este objetivo.

Para este proyecto vamos a tratar de escoger la que mejor se adecúe al efecto que queremos encontrar: *una manera cómoda de trabajar en comunidad.*

Las herramientas más conocidas de virtualización de máquinas son Oracle VM VirtualBox, VMWare y Docker, aun que existen algunas diferencias entre ellos que trataremos a continuación.

2.2.2 Análisis de las herramientas

Vamos a ver qué aporta cada una de las herramientas para tomar una decisión.

Oracle VM VirtualBox Es un producto de potente virtualización x86 and AMD64/Intel64 tanto para empresas como para uso doméstico.

VirtualBox no solo aporta ricas características, un producto de alto rendimiento para los clientes de empresas si no que es la única solución libremente disponible como Software de código abierto bajo la licencia GNU/GPL v2.

A pesar de que nos aporta la posibilidad de uso libre de la aplicación, esta herramienta necesita virtualizar una máquina entera para su funcionamiento y una configuración que no permite que las máquinas se compartan con facilidad.

VMWare Es un producto con características parecidas a las de VirtualBox salvo que no está protegido con una licencia libre.

Nos encontramos con problemas parecidos a los anteriores, la compartición de máquinas entre desarrolladores se vuelve lenta y pesada y a demás esta herramienta no nos permite uso completo sin licencia.

Docker Es un sistema de virtualización de máquinas agrupadas en contenedores.

Permite virtualizar una o varias máquinas virtuales a partir de un fichero de texto plano, aprovecha los recursos que sean compartidos entre máquinas y anfitriones para reducir el consumo y almacenamiento y está protegido bajo licencia Apache 2.0.

- Nos permite definir exactamente las mismas características a virtualizar para todos los desarrolladores.
- Permite una compartición de máquinas homogénea y ligera (sólo un archivo de texto).
- Reduce considerablemente el consumo de la arquitectura a virtualizar.
- Licencia Apache 2.0

2.2.3 Conclusiones

Nos encontramos ante dos problemas importantes a la hora de desarrollar un proyecto de manera colaborativa. Estos problemas son: cómo **organizarse** para que todos los desarrolladores trabajen con las mismas herramientas de desarrollo, en las mismas **versiones**, sobre el mismo sistema operativo; y cómo facilitar que la herramienta funcione en la mayoría de equipos, independientemente de que el desarrollador tenga que **desplegar la arquitectura** y que su máquina no sea tan potente como para lanzar todo lo necesario y ponerse a colaborar.

Para mejorar la homogenidad entre distribuciones, kernels, versiones... y la comunicación entre desarrolladores y facilitarles la tarea lo más posible, elegiré **Docker** como herramienta para virtualizar las máquinas y desplegar

el proyecto que se encargarán de servir Kapi Solutions.

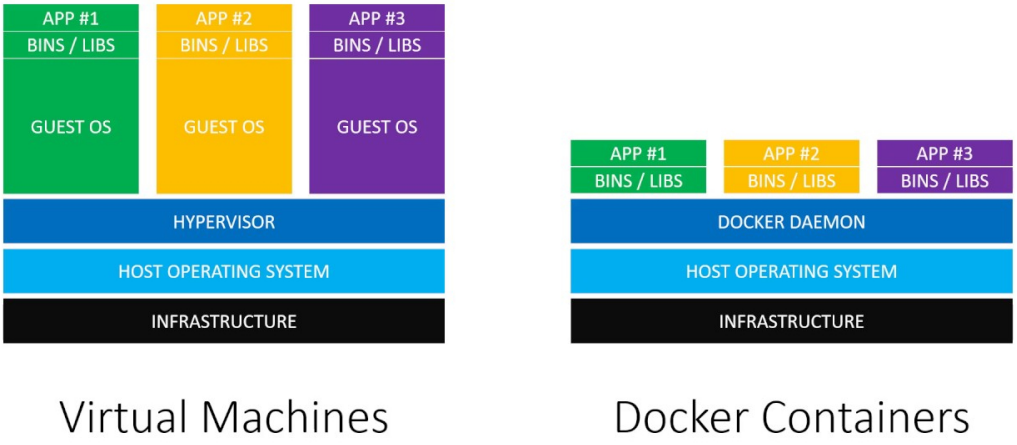
Otro problema que resolvemos a la hora de implementar el despliegue de desarrollo con **docker**, es que a la hora de desplegar el entorno de producción, podemos hacerlo con la misma arquitectura que se ha utilizado en el entorno de desarrollo utilizando herramientas de cloud como ofrecen distintas compañías de arquitecturas en la nube que permiten despliegues de docker. Una de ellas sería AWS ECS, lo que nos permitiría llevar un desarrollo con integración contínua [1].

2.3 Integración de Kapi en Docker

Esta herramienta permite virtualizar máquinas, instalar servicios en las máquinas, configuración de estos servicios y de los puertos expuestos por la máquina entre otras funcionalidades y todas ellas únicamente usando un fichero descriptor del sistema que vayamos a lanzar que prepara, descarga y lanza las máquinas que describamos.

Queremos preparar dos máquinas que sirvan la aplicación, una que expondrá el servicio apache y otra expondrá la base de datos.

Figura 1: Comparativa de recursos VM vs Docker



La figura1 muestra un esquema visual para entender el funcionamiento de Docker, este no necesita la instalación y configuración de un sistema opera-

[1] Integración Contínua (CI): sistema atomatizado para construir, desplegar y testear tu aplicación con cada git push.

tivo completo para lanzar las máquinas. Estas aprovechan todos los recursos des sistema anfitrión haciendo que el espacio y el rendimiento mejoren.

¿Qué es lo que necesitamos? Para soportar nuestra aplicación deberemos configurar dos contenedores Docker; los dos visibles entre sí y uno de ellos expuesto al exterior para dar el servicio.

El primero de los dos soportará la carga de apache y PHP mientras que el segundo le sirve la base de datos. Separados conseguimos mayor seguridad.

Para preparar las imágenes que usaremos en los contenedores, intentaremos priorizar en imágenes oficiales con modificaciones de configuración adaptadas al proyecto. De esta manera, la descarga de las mismas será más cómoda para el desarrollador. Elegiremos las versiones para las dos imágenes.

Imagen MySQL Para la base de datos MySQL usaremos una imagen oficial con el servicio MariaDB (un fork de MySQL completamente libre y abierto).

Imagen Apache y PHP Para el servicio Apache y el intérprete PHP usaremos una imagen oficial de la comunidad que aporta estas características.

Imagen PHPMyAdmin Utilizaremos un tercer contenedor virtualizado que se conectará al de la base de datos para servir estos datos con PHPMyAdmin y así poder gestionar la base de datos de una manera cómoda.

Comunicación entre máquinas Las dos imágenes servirán para lanzar los dos contenedores Docker que usaremos. Estos deben tener comunicación entre ellos para servir la base de datos y el que sirve la web, expondrá el servicio Apache al exterior.

Para ello configuraremos y decidiremos los puertos expuestos.

Configuración de puertos de los contenedores Vamos a definir los puertos y la configuración de los mismos para cada contenedor.

Configuración de puertos del contenedor Apache + PHP La tabla muestra los puertos de Docker que se exponen en red local, los que expone la máquina anfitrión y los protocolos utilizados en la comunicación entre el contenedor **apache** y el **cliente**.

Tabla 2: Puertos de la máquina Apache + PHP

Puerto Docker	Puerto externo	TCP/UDP	Protocol
80	80	TCP	http
443	443	TCP	https

Configuración de puertos del contenedor MySQL La tabla muestra los puertos de Docker que se exponen en red local, los que expone la máquina anfitrión y los protocolos utilizados en la comunicación entre el contenedor de la **base de datos** y el contenedor **apache**.

Tabla 3: Puertos de la máquina MySQL

Puerto Docker	Puerto externo	TCP/UDP	Protocol
3306	–	TCP	sql

Configuración de puertos del contenedor PHPMyAdmin El contenedor de PHPMyAdmin, expondrá el puerto 80 redirigido al 8080 de la máquina anfitrión, esto se observa en la tabla.

Tabla 4: Puertos de la máquina PHPMyAdmin

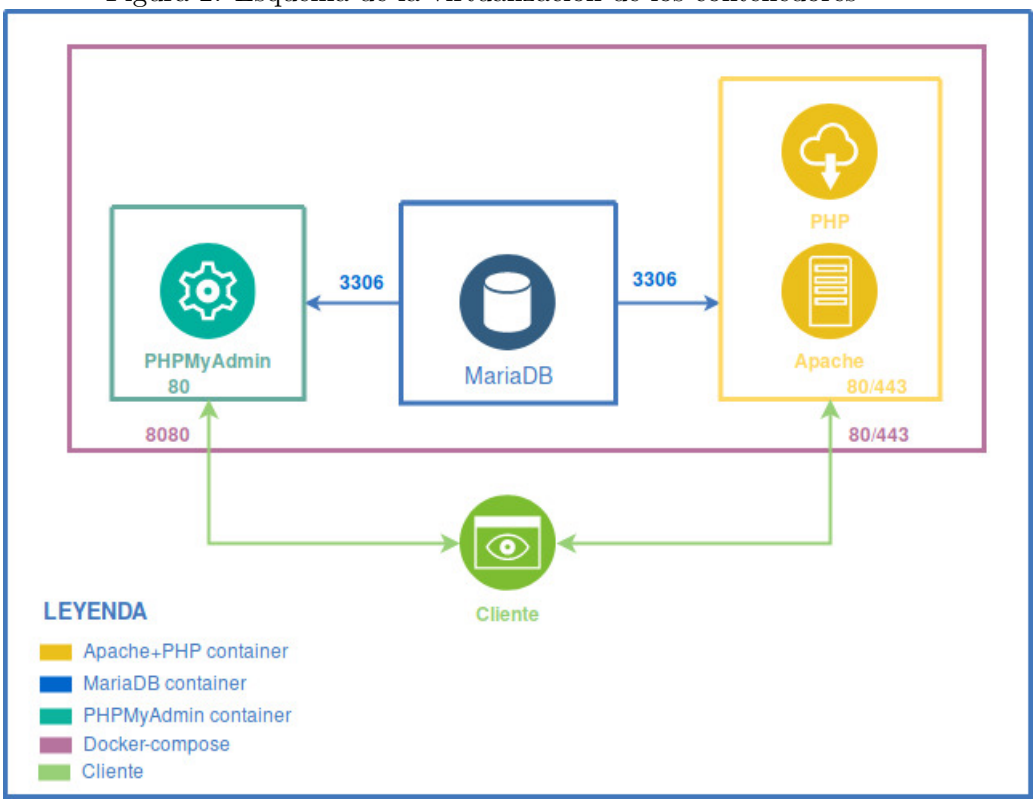
Puerto Docker	Puerto externo	TCP/UDP	Protocol
80	8080	TCP	http

De esta manera, sólo expondremos exterior los puertos Apache 80 y 443 para http y https respectivamente, quedando el contenedor MySQL y por tanto la base de datos, comunicada con el contenedor Apache en red local pero sin exponer ningún puerto a internet.

En la imagen se muestra cómo queremos organizar los contenedores virtualizando los servicios.

Configuración de variables de entorno de los contenedores El sistema docker permite montar archivos en los contenedores así que montaremos un volumen en la ruta `/app` del servidor Apache con todo el proyecto y en el contenedor de MariaDB, volcamos un script `.sql` con los datos de ejemplo para poder trabajar en el proyecto

Figura 2: Esquema de la virtualización de los contenedores



Esto se hace automáticamente al lanzar los contenedores y permite que el desarrollador trabaje en su directorio local y vea los cambios reflejados en los contenedores virtualizados.

2.3.1 Infraestructura de desarrollo

Una vez tomadas las decisiones para realizar la virtualización del sistema, utilizamos *docker* con *docker-compose*. Estas tecnologías nos permitirán virtualizar tres máquinas, cada una con su especificación concreta, todas ellas en un solo fichero de texto plano. Esto nos permite que el montaje de las máquinas sea ligero y automático.

Para iniciar la virtualización necesitamos docker y docker-compose en nuestra máquina, y ejecutar el fichero **docker-compose.yml** 1 con el siguiente comando en la Terminal/Consola:

```
$ docker-compose -f docker-compose.yml up
```

Código 1: docker-compose.yml

```
version: '2'
services:
  mariadb:
    image: mariadb
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=kapi
      - MYSQL_DATABASE=kapi
      - MYSQL_USER=kapi
      - MYSQL_PASSWORD=kapi
      - MYSQL_ROOT_HOST
    ports:
      - '3306:3306'
    volumes:
      - './doc/populate:/docker-entrypoint-initdb.d/'
  php-apache-dev:
    image: webdevops/php-apache-dev
    ports:
      - '80:80'
      - '443:443'
    volumes:
      - '.:/app'
  phpmyadmin:
    image: phpmyadmin/phpmyadmin
    environment:
      - PMA_HOST=mariadb
      - PMA_USER=kapi
      - PMA_PASSWORD=kapi
    ports:
      - '8080:80'
```

Al ejecutar esta configuración de docker, tendremos tres máquinas virtualizadas con los servicios MariaDB para la base de datos, Apache para servir la web y PHPMyAdmin para gestionar la base de datos.

El consumo de recursos producido por el sistema completo es el que se especifica en la figura 3.

Figura 3: Consumo de recursos de los contenedores

Containers: **25** Running: **3** CPU: **0.166 %** MEMORY USAGE: **251.13 MB**

Name	CONTAINER	CPU	MEM USAGE / LIMIT	NET I/O	BLOCK I/O
kapisolution_mariadb_1	150bac85c182	0.064 %	102.45 MB/ 3.86 GB	281.98 KB/ 545.03 KB	8.65 MB/ 2.32 MB
kapisolution_php-apache-dev_1	9e5f5af0b9ab	0.076 %	90.63 MB/ 3.86 GB	144.76 KB/ 3.12 MB	5.77 MB/ 112 KB
kapisolution_phpmyadmin_1	6bcbb9f20f46	0.031 %	58.05 MB/ 3.86 GB	608.44 KB/ 668.2 KB	5.75 MB/ 164 KB

3 Implementación de la aplicación web

En esta sección se tratarán en detalle las tareas de implementación de la aplicación web por mi parte. Lo que serán las tareas de back-end de despliegue de árbol de directorios que utilizaremos en nuestra herramienta, el diseño y configuración de la base de datos y mi contribución en código php sobre la aplicación.

3.1 Árbol de directorios

Decidir cómo estructurar el árbol de directorios, es una tarea muy importante y más aún en nuestro caso.

Para definir claramente las partes del proyecto en **backend** y **frontend**, para ello usaremos dos carpetas con dichos nombres (backend y frontend).

En la carpeta *frontend*, tendremos todos los archivos `.php` que se encarguen de pintar el html con los datos, por otro lado tendremos en la carpeta *backend* todos los archivos `.php` que trabajen con la base de datos, así como realizar la conexión y desconexión de la base de datos, todas las consultas para preparar los datos que se pintan en la web, todas las consultas para actualizar, insertar o eliminar contenido ya existente (tanto de artículos nuevos, existentes, preguntas, votaciones así como de los datos de usuario).

La carpeta *files* almacenará los archivos estáticos del servidor (imágenes de usuarios, imágenes de los artículos y todos aquellos archivos que sean necesarios para la web Ej. iconos, fuentes), y por último dos carpetas más con los archivos `.js` para las funciones javascript y una carpeta con los archivos `.css` que decorarán la página. Para consolidar las rutas, crearemos un archivo `.php` en la raíz del proyecto con el nombre de la ruta que queramos acceder (*index*, *modificaciones*, *login*, *logout*, *creararticulo*, *comunidad*...)

El árbol de desarrollo se puede observar en la figura 4 **árbol de directorios**

3.2 Diseño de la base de datos

En esta sección se especifica cómo está diseñada la base de datos: sus tablas y sus relaciones. El esquema de la base de datos se puede observar en la figura 5.

3.3 Contribuciones sobre el backend

Como he mencionado, dentro de la plataforma, me he encargado principalmente de la parte del backend. Las partes del backend más significativas que he desarrollado son las siguientes:

Figura 4: Árbol de directorios

```
/
├── ruta.php (index, articulos, login...)
├── backend
│   └── archivos backend.php
├── frontend
│   └── archivos frontend.php
├── files
│   ├── fonts
│   │   └── archivos de fuentes
│   ├── icons
│   │   └── paquete de iconos
│   └── img
│       ├── articulos
│       │   └── imágenes asociadas a los artículos
│       ├── rol
│       │   └── imágenes asociadas a los roles
│       └── usuario
│           └── imágenes asociadas al usuario por ID
├── css
│   └── archivos .css
└── js
    └── archivos .js
```

3.3.1 Conexión y desconexión de la base de datos

En el fichero `conexion.php` 2 se programa la conexión a la base de datos para la configuración especificada en la virtualización, esta especificación es una IP local, el nombre de la base de datos, el usuario y la contraseña, estos datos están especificados en la figura 1.

Código 2: backend/conexion.php

```
$con = mysqli_connect($servername, $username, $password);
```

En el siguiente fichero `desconexion.php` 3, se cierra la conexión con el servidor para dejar de enviar peticiones.

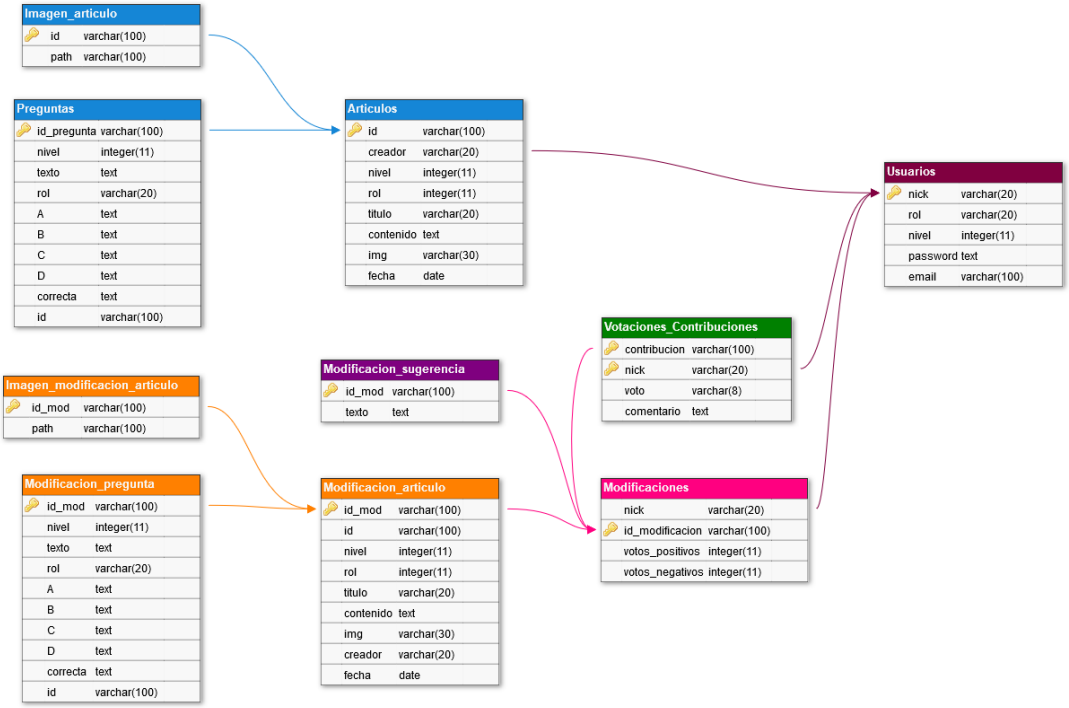
Código 3: backend/desconexion.php

```
$con->close();
```

3.3.2 Búsqueda en la aplicación

En el siguiente código de la figura4 cargamos todos los archivos que se encargarán de realizar una búsqueda en la base de datos (artículos5, usuarios8,

Figura 5: Esquema relacional de las entidades de la base de datos



creaciones6 y modificaciones7).

Código 4: backend/busqueda.php

```
include 'busqueda_articulos.php';
include 'busqueda_usuarios.php';
include 'busqueda_creaciones.php';
include 'busqueda_modificaciones.php';
```

En el fichero `backend/busqueda_artículos.php` 5 se detalla la consulta de búsqueda de artículos.

Código 5: backend/busquedaarticulos.php

```
$sql ='SELECT * FROM Articulos WHERE titulo LIKE "%'.
    $_GET["search"] .'%" AND contenido LIKE "%'.$_GET
    ["search"] .'%" ORDER BY nivel, rol;';
```

En el fichero `backend/busqueda_creaciones.php` 6 se hace una consulta sobre la tabla modificaciones, a las que sean nuevas (se considera una creación).

Código 6: backend/busquedacreaciones.php

```
$sql ='SELECT␣*␣FROM␣Modificacion_articulo␣␣a,␣
    Modificaciones␣m␣WHERE␣a.id␣=␣0␣AND␣a.id_mod␣=␣m.
    id_modificacion␣AND␣a.titulo␣LIKE␣"%'.$_GET["
    search"] .'%"␣AND␣a.contenido␣LIKE␣"%'.$_GET["
    search"] .'%"␣ORDER␣BY␣a.nivel,␣a.rol;';
```

En el fichero `backend/busqueda_modificaciones.php` 7, al contrario que las creaciones, se hace una petición a aquellas modificaciones que sí vienen de un artículo existente, esto se considera una modificación.

Código 7: backend/busquedamodificaciones.php

```
$sql ='SELECT␣*␣FROM␣Modificacion_articulo␣␣a,␣
    Modificaciones␣m␣WHERE␣a.id␣<>␣0␣AND␣a.id_mod␣=␣m
    .id_modificacion␣AND␣a.titulo␣LIKE␣"%'.$_GET["
    search"] .'%"␣AND␣a.contenido␣LIKE␣"%'.$_GET["
    search"] .'%"␣ORDER␣BY␣a.nivel,␣a.rol;';
```

En el fichero `backend/busqueda_usuarios.php` 8, se especifica la consulta utilizada para la obtención de los usuarios que cumplan con la búsqueda.

Código 8: bakend/busquedausuarios.php

```
$sql ='SELECT␣*␣FROM␣Usuarios␣WHERE␣nick␣LIKE␣"%'.
    $_GET["search"].'%";';
```

En el fichero `backend/busqueda_preguntas.php` 9, se hace la petición a la base de datos de las preguntas que cumplen con la búsqueda realizada.

Código 9: backend/busquedapreguntas.php

```
$sql ='SELECT␣*␣FROM␣Preguntas␣WHERE␣texto␣LIKE␣"%'.
    $_GET["search"] .'%";';
```

Estas consultas devuelven al *frontend* todos los datos necesarios para pintar los resultados de búsqueda.

3.3.3 Mostrar creación de un artículo

A la hora de mostrar una creación [2] de artículo, hago una petición a la base de datos con la fila de dicho artículos, todas las preguntas que lo componen y los artículos relacionados con este por nivel y rol. Para esto, queda el código[3] de la figura 10.

[2] Una creación, es un artículo recién creado, esto puede venir de un artículo nuevo, o bien de un artículo que se haya modificado a partir de uno existente.

[3] Una programación parecida es la que se utiliza para mostrar una modificación.

Código 10: backend/creacion.php

```
$sql ="SELECT␣*␣FROM␣Modificacion_articulo␣a,␣
    Modificaciones␣m␣WHERE␣a.id_mod␣=␣'".$id."'␣AND␣a
    .id_mod␣=␣m.id_modificacion;";
```

3.3.4 Mostrar artículo publicado

El archivo `articulo.php` 11 devuelve los datos asociados al artículo que se desea especificado en el parámetro *id* de la url.

Código 11: backend/articulo.php

```
$sql ="SELECT␣*␣FROM␣Articulos␣WHERE␣id␣=␣'".$id."';"
    ;
```

3.3.5 Crear un artículo

El archivo `backend/creararticulo.php` 12 se ejecutará después de rellenar el formulario de creación de un artículo (ya sea una creación o una modificación). Se encargará de recoger los datos del formulario, insertar el artículo con sus respectivas relaciones y subir la imagen asociada al artículo en su localización correcta y con un formato accesible y termina volviendo a la página principal.

Código 12: backend/creararticulo.php

```
require("recogeDatosArticulo.php");
require("insertaArticulo.php");
$total = count($_FILES['imagen']['name']);
require("subeImgArticulo.php");
```

Para subir las imágenes asociadas a los artículos, las almacenaremos siempre en una ruta especificada por el *id* de la modificación, de esta manera podremos almacenar tantas imágenes como queramos en nuestro sistema, sin colisión de nombres y de manera controlada.

Código 13: backend/subeImgArticulo.php

```
$ficheroAsubir = $_FILES['imagen']['tmp_name']
    ][$i];
$directorio ="../files/img/articulos/".
    $id_mod . "/" ;
$nombreImagen = $i+1 . ".jpg";
$fichero_subido = $directorio . $nombreImagen
    ;
if (!file_exists($directorio)) {
```

```
        mkdir($directorio, 0777, true);  //crea
            el directorio si no existe
    }
    if (move_uploaded_file($ficheroAsubir,
        $fichero_subido)) {
        echo "El fichero es valido y se subio con
            exito.\n";
        $sql = "INSERT INTO
            Imagen_modificacion_articulo (id_mod,
            path) VALUES ('$id_mod','
            $nombreImagen');";
        $consulta = mysqli_query($con, $sql) or
            die("No se pudo insertar la imagen");
```

3.3.6 Generar exámenes

EL archivo `backend/generaexamen.php`14 necesita de dos parámetros; *nivel* y *rol*, a partir de estos, ejecuta una consulta sobre la base de datos que obtiene 10 preguntas y sus respuestas de manera aleatoria, correspondiendo al patrón de nivel y rol.

Código 14: backend/generaexamen.php

```
$sql ='SELECT * FROM Preguntas WHERE nivel = '.
    $nivel.' AND rol = "'.$rol.'" ORDER BY RAND()
    LIMIT 10;';
```

3.3.7 Guardar examen y nota

El examen que realiza el usuario, es un formulario que envía la nota a `backend/guardarExamen.php` 15, este se encargará de guardar el nivel y el rol del examen, generando un ID para el examen que insertamos en la tabla *Realiza_Examen* el usuario, el examen y la nota obtenida.

Código 15: backend/guardarExamen.php

```
$sql = "INSERT INTO Examenes (nivel, rol)
    VALUES ('$nivel','$rol');";
$consulta = mysqli_query($con, $sql) or die("
    No se pudo insertar el rol y el nivel");

$id_ex = mysqli_insert_id($con);
$sql = "INSERT INTO Realiza_Examen (nick,
    id_ex, nota) VALUES ('$nick', '$id_ex', '
    $nota');";
```

3.3.8 Información en el perfil de usuario

En la pantalla del perfil de usuario, se muestran los datos del usuario, así como los artículos que han sido publicados por ellos, los que han sido creados por ellos y los que han modificado. Para ello se utilizan los ficheros `backend/creaciones_usuario.php`17, `backend/modificaciones_usuario.php`18, `backend/publicaciones_usuario.php`19 y `backend/usuario.php`16 que se encargan de recoger los datos que se pintarán en la web.

Código 16: backend/usuario.php

```
$id =$_GET["id"];
$id = stripcslashes($id);
$id = mysql_real_scape_string($id);
$sql ='SELECT␣*␣FROM␣Usuarios␣WHERE␣nick␣=␣"'. $id .'
    ";';
$consulta = mysqli_query($con, $sql) or die("Error␣en
    ␣consulta␣sobre␣la␣tabla␣Usuarios");
$usuario = mysqli_fetch_array($consulta);

require 'creaciones_usuario.php';
require 'modificaciones_usuario.php';
require 'publicaciones_usuario.php';
```

Código 17: backend/creacionesusuario.php

```
$sql ='SELECT␣*␣FROM␣Modificacion_articulo␣a,␣
    Modificaciones␣m␣WHERE␣a.id␣=␣0␣AND␣a.id_mod␣
    =␣m.id_modificacion␣AND␣a.creador␣=␣"'. $_GET
    ["id"] .'"␣ORDER␣BY␣fecha␣DESC;';
```

Código 18: backend/modificacionesusuario.php

```
$sql ='SELECT␣*␣FROM␣Modificacion_articulo␣a,␣
    Modificaciones␣m␣WHERE␣a.id␣<>␣0␣AND␣a.id_mod
    ␣=␣m.id_modificacion␣AND␣a.creador␣=␣"'.
    $_GET["id"] .'"␣ORDER␣BY␣fecha␣DESC;';
```

Código 19: backend/publicacionesusuario.php

```
$sql ='SELECT␣*␣FROM␣Articulos␣WHERE␣creador␣=␣"'
    . $_GET["id"] .'"␣ORDER␣BY␣fecha␣DESC;';
```

3.3.9 Hasheo de la contraseña

Para proteger al usuario, almacenamos las contraseñas cifradas con md5, para ello las hasheamos en el registro 20 y en el inicio de sesión 21.

Código 20: backend/signin.php

```
$passdb = md5($password);
```

Código 21: backend/login.php

```
if($passdb == md5($passIntroducida)){
```

3.4 Población masiva de datos

Para poder tener una visión realista de nuestra web tanto en diseño como a la hora de comprobar las consultas, necesitamos de una población de datos sobre la base de datos.

He utilizado una herramienta de generación de datos, exportando los resultados en `CSV` [4], de esta manera y siguiendo bien la definición de los datos de la base de datos, podemos poblarla con datos (algunos aleatorios, otros concretos) y tener una idea clara durante el desarrollo de si los resultados que se obtienen son correctos.

4 Securización de la aplicación

En esta sección trataremos tanto el análisis realizado a la herramienta, como la resolución de los problemas de seguridad que se han encontrado.

Para realizar el análisis he utilizado la herramienta **ZAP** de **OWASP**, por ser una herramienta dedicada al análisis de vulnerabilidades de una aplicación web completamente abierta, desarrollada por Open Web Application Security Proyect, lo que nos aporta un análisis imparcial y con una herramienta de código abierto.
Esta herramienta, ZAP, la ejecutaremos desde una máquina virtual con Kali Linux y OWASP ZAP debidamente configurado, tanto las tarjetas red, como las ip de las máquinas y los contenedores.

Una vez configuramos la máquina virtual, desplegamos la aplicación web, y procedemos a atacarla en un entorno local seguro.

4.1 Análisis de vulnerabilidades

Utilizando la herramienta VirtualBox, virtualizaremos una máquina Kali (proporcionada por el director del TFG), atacaremos nuestro sistema con **OWASP ZAP** para obtener un análisis de las vulnerabilidades detectadas en nuestro sistema y reportar un informe detallado de las más importantes

[4] CSV: comma-separated values, es un formato de datos simple en el que los datos están separados por comas.

y directrices de cómo corregirlas.

El informe que reporta OWASP refleja las siguientes vulnerabilidades ordenadas por riesgo:

- **Alta**
  - Cross Site Scripting (*reflejada*)
  - Falla por inyección SQL (4)
- **Media**
  - Exploración de directorios (7)
  - X-Frame-Options Header Not Set (37)
- **Baja**
  - Cookie No HttpOnly Flag
  - Cross-Domain JavaScript Source File Inclusion (73)
  - Password Autocomplete in Browser (8)
  - Web Browser XSS Protection Not Enabled (41)
  - X-Content-Type-Options Header Missing (57)

Esta lista ha sido obtenida del análisis como puede verse en la figura 18. Lo que hace un total de **227** vulnerabilidades encontradas.

Trataré de resolver el mayor número de vulnerabilidades encontradas por la herramienta, priorizando por nivel de riesgo y por número de apariciones, siendo lo más prioritario **falla por inyección SQL**, **X-Frame-Options Header Not Set** y **Cross-Domain JavaScript Source File Inclusion**.

De estas vulnerabilidades, pertenecen al **back-end** las **cuatro** detectadas como *falla por inyección SQL*, las **siete** que corresponden a la *exploración de directorios* y las **treinta y siete** amenazas detectadas como *X-Frame-Options Header Not Set*, lo que suponen **cuarenta y ocho** vulnerabilidades en el **back-end** de las doscientas veintisiete, el resto corresponden al desarrollo del **front-end**.

4.2 Vulnerabilidades resueltas

A partir de las vulnerabilidades que he encontrado en nuestra aplicación, resolveré algunas importantes.

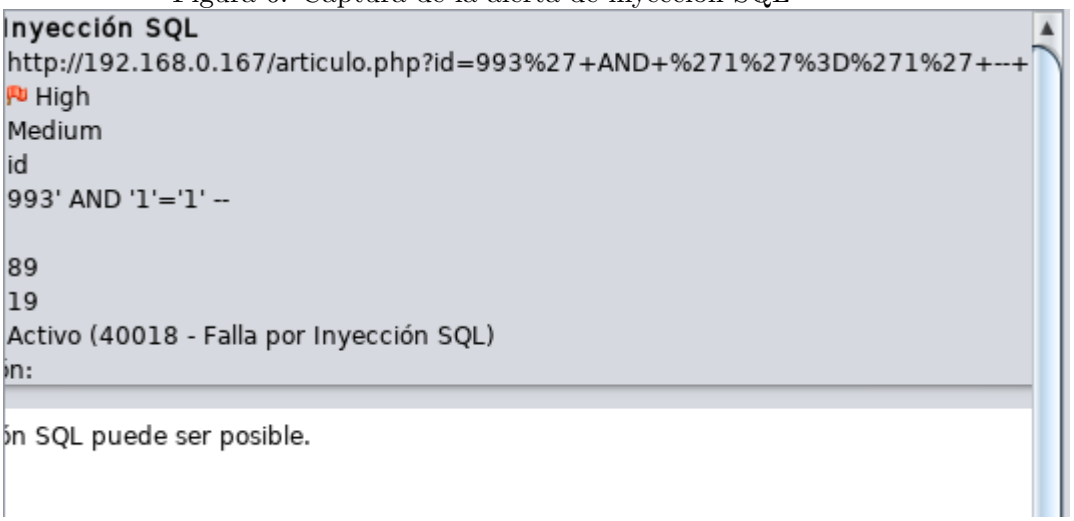
Las vulnerabilidades resueltas son las siguientes:

4.2.1 Falla por inyección de SQL

La inyección de SQL es una de las vulnerabilidades web más explotadas hasta la fecha. Por eso, es una de las más importantes a resolver.

Las cuatro encontradas, son del mismo tipo, *'AND'1'='1'–*, como se observa en la figura 6.
En nuestro caso, vamos a resolver el problema sobre articulo.php.

Figura 6: Captura de la alerta de inyección SQL



Código 22: SQL inyetion vulnerable

```
$sql ="SELECT␣*␣FROM␣Articulos␣WHERE␣id␣=␣'".$_GET["
    id"]."';";
```

Como se puede observar en el código 23, se ejecuta la query con el valor de la llamada directamente, para solucionarlo debemos recoger en una variable los datos, como muestra el código 22, y escaparlos debidamente para eliminar código malicioso.

Código 23: SQL inyection resuelto

```
$id =$_GET["id"];
$id = stripcslashes($id);
$id = mysql_real_scape_string($id);
$sql ="SELECT␣*␣FROM␣Articulos␣WHERE␣id␣=␣'".$id."';"
    ;
```

Realizando esta modificación sobre los otros cuatro ficheros que presentan la amenaza, quedaría resuelta en todos los puntos que se localizó.

Con esto quedan resueltas **cuatro** (4) vulnerabilidades de riesgo **alto**.

El resto de zonas solucionadas son las siguientes 24, 25,26, 27,28, 29.

Código 24: SQL inyetion vulnerable

```
$sql ="SELECT␣*␣FROM␣Modificacion_articulo␣a,␣
    Modificaciones␣m␣WHERE␣a.id_mod␣=␣'".$_GET["id"].
    "'␣AND␣a.id_mod␣=␣m.id_modificacion;";
```

Código 25: SQL inyection resuelto

```
$id =$_GET["id"];
$id = stripcslashes($id);
$id = mysql_real_scape_string($id);
$sql ="SELECT␣*␣FROM␣Modificacion_articulo␣a,␣
    Modificaciones␣m␣WHERE␣a.id_mod␣=␣'".$id."'␣AND␣a
    .id_mod␣=␣m.id_modificacion;";
```

Código 26: SQL inyetion vulnerable

```
$sql ="SELECT␣*␣FROM␣Modificacion_articulo␣a,␣
    Modificaciones␣m␣WHERE␣a.id␣<>␣0␣AND␣a.id_mod␣=␣'
    ".$_GET["id"] ."'␣AND␣a.id_mod␣=␣m.
    id_modificacion␣ORDER␣BY␣fecha␣DESC;";
```

Código 27: SQL inyection resuelto

```
$id =$_GET["id"];
$id = stripcslashes($id);
$id = mysql_real_scape_string($id);
$sql ="SELECT␣*␣FROM␣Modificacion_articulo␣a,␣
    Modificaciones␣m␣WHERE␣a.id␣<>␣0␣AND␣a.id_mod␣=␣'
    ".$id ."'␣AND␣a.id_mod␣=␣m.id_modificacion␣ORDER␣
    BY␣fecha␣DESC;";
```

Código 28: SQL inyetion vulnerable

```
$sql ='SELECT␣*␣FROM␣Usuarios␣WHERE␣nick␣=␣"'. $_GET[
    "id"] .'";';
```

Código 29: SQL inyection resuelto

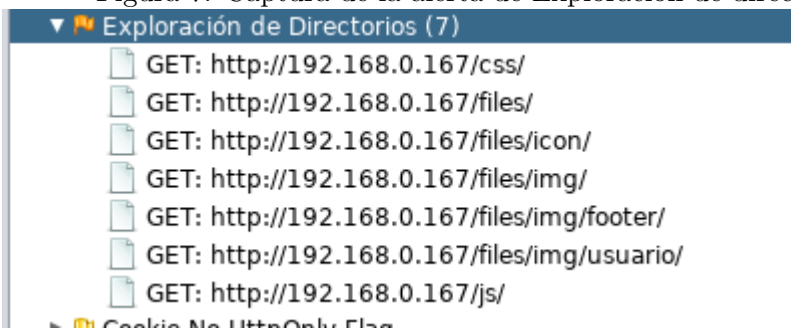
```
$id =$_GET["id"];
$id = stripcslashes($id);
$id = mysql_real_scape_string($id);
$sql ='SELECT␣*␣FROM␣Usuarios␣WHERE␣nick␣=␣"'. $id .'
    ";';
```

4.2.2 Exploración de directorios

En la figura 7, se listan los directorios que se han detectado accesibles, estos deben serlo para poder servir tanto los archivos CSS, el paquete de iconos,

las imágenes del footer[5], las imágenes del perfil de los usuarios y los archivos que JavaScript que se ejecutan en el cliente, por tanto tambien deben tener acceso y ser descargados.

Figura 7: Captura de la alerta de Exploración de directorios



Se puede observar que el ataque no localiza las carpetas *backend* y *frontend* así como la carpeta *docs*, lo cual no supone una amenaza.

4.2.3 X-Frame-Options Header Not Set

Esta alerta, nos indica que en las llamadas del servidor, no estamos solicitando una cabecera que especifica el ámbito de las llamadas que vamos a permitir o no.

En nuestro caso queremos que todas las llamadas sean internas por lo que incluiremos la cabecera *'X-Frame-Options: SAMEORIGIN'*.

Código 30: X-Frame-Option Header Not Set resuelto

```
<?php
  session_start();
  header('X-Frame-Options: SAMEORIGIN');
```

Así resolvemos la vulnerabilidad incluyendo esta cabecera en los archivos de la raíz del proyecto.

Con esto quedan resueltas **treinta y siete** (37) vulnerabilidades de riesgo **medio**.

4.2.4 Cross-Domain JavaScript Source File Inclusion

Como se puede ver en el código 31, las librerías de Bootstrap y jQuery, se cargan desde un archivo de dominio externo.

[5]Footer: pie de página de la web, contiene imágenes que deben ser accesibles.

Para eso, modificaré los archivos de **front-end** que contengan tales vulnerabilidades. Descargando las librerías necesarias, guardándolas en la carpeta /js creada para tal efecto y linkandola en cada una de las páginas que contengan la vulnerabilidad.

Código 31: Cross-Domain JavaScript Source File Inclusion

```
<script src="https://ajax.googleapis.com/ajax/libs/
    jquery/3.3.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/
    bootstrap/3.3.7/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="https://maxcdn.
    bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.
    min.css">
```

Código 32: Cross-Domain JavaScript Source File Inclusion resuelto

```
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<link rel="stylesheet" href="css/bootstrap.min.css">
```

Importando debidamente todas las librerías necesarias, almacenándolas en local para que la carga sea desde el propio servidor de la aplicación, quedan resueltas **setenta y tres** (73) vulnerabilidades de riesgo **bajo**.

5 Resultados y discusión

En esta sección vamos a analizar cómo cumple la aplicación con los objetivos planteados para el proyecto. Veremos qué casos quedan cubiertos en el uso de la aplicación como usuario de la misma, aquellos que hemos cubierto para el desarrollador y las tareas que pueden plantearse para el futuro, como posibles ampliaciones a realizar por el equipo de desarrollo.

5.1 Uso de la aplicación web

A continuación se detalla y explica cómo usar nuestra aplicación. Estas explicaciones se acompañan de capturas de las distintas pantallas de la aplicación.

En la figura 8 se muestra un formulario para el registro del usuario, una lista con los artículos de nuestra aplicación y la barra superior de la aplicación, esta barra nos permite acceder al buscador de artículos, para mostrarlo a cualquier usuario [6]. Arriba a la derecha, se encuentra la zona de la sesión, como aún no se ha iniciado sesión, se muestra un botón rojo

[6]Incluidos aquellos ususarios que no se han registrado o no han iniciado sesión.

para acceder a la pantalla de login.

En la figura 9 se muestra un formulario con los campos *usuario* y *contraseña*, enviando estos campos al backend, el que comprueba los datos con los de la base de datos.

En esta vista 10, se listan los artículos ya publicados, pudiendo ordenar los mismos por fecha o por nivel.

En la figura 11, muestra los artículos que son una modificación de otro artículo ya publicado pero esta modificación aún no ha sido publicada y se encuentran en fase de votación, junto con una barra de porcentaje de votaciones ordenados por fecha de creación.

Este orden se puede modificar y ordenarlo por nivel de mayor a menor o al contrario.

La figura 12, de forma parecida a la figura 11, muestra los artículos que han sido creados, esta vez desde cero y sin corresponder a uno ya publicado con anterioridad, y que están listos para votaciones.

Al acceder a una de las modificaciones, pasamos a la vista de la figura 14, en esta vista, podemos ver la información del artículo y modificarlo si cumplimos con los requisitos de rol y nivel.

El usuario puede crear artículos o modificarlos. Para esto, puede llegar a la vista de la figura 13 o bien creando uno nuevo o modificando uno existente[7], aquí el usuario completará tanto el título como el contenido y las preguntas del artículo. Al subir la creación, esta se visualizará en su vista correspondiente (11 o 12) y estará preparada para recibir votaciones.

En la figura 15 se muestra la información del perfil de usuario así como la posibilidad de editarlo si fuese el usuario de la sesión.

La información que se muestra en el perfil son los datos de usuario, las modificaciones y creaciones que ha hecho y los artículos publicados por el usuario que se está solicitando.

En la vista de pruebas de nivel 16, veremos un listado con las pruebas realizadas por el usuario, con la posibilidad de realizar la prueba del nivel inmediatamente superior al del usuario.

[7]en este caso, se rellenarán automáticamente los campos del formulario con los que había en el original

Por último, en la vista de examen 17, se muestran las preguntas para resolver el examen del nivel y rol que corresponda al usuario. Cada pregunta con sus cuatro posibles respuestas y un pequeño formulario para recoger la respuesta del usuario.

5.2 Uso como desarrollador

El desarrollador dispone de **git/GitHub** y de **docker** para contribuir en la plataforma. A continuación describo cómo colaborar con nosotros.

El repositorio de github que utilizamos es el siguiente:

`https://github.com/kapisolution/kapi_solution.git`

Para trabajar con git sin problemas de mezclas, implementaremos la metodología de trabajo *git-flow*, manteniendo la rama **master** aislada del desarrollo hasta su integración y pruebas en la rama **development**, a partir de esta rama de desarrollo, cada desarrollador creará una rama para su trabajo y cuando haya probado sus cambios en su rama, **mergeará** con **development** y probará sus cambios en esta rama, junto con todas las novedades que haya en la rama.

Por último, una vez probado todo en **development**, se podrá hacer un *merge* a la rama **master**, integrando los cambios en la plataforma final.

Usando esta técnica de trabajo, nunca debemos permitir que la rama **master** se ponga por delante de la rama **development**.

El desarrollador se ayudará de **docker** y en concreto de **docker-compose** 1 para virtualizar el entorno de desarrollo de la plataforma.

Para comenzar con el desarrollo del proyecto, partimos del proyecto original con:

```
$ git clone https://github.com/kapisolution/kapi_solution.git
```

La configuración de la virtualización del entorno de pruebas, necesita la instalación de `docker` y `docker-compose`, para lanzarlo, ejecuatmos el siguiente comando:

```
$ docker-compose -f docker-compose.yml up
```

Esto virtualizará los tres contenedores, montará la carpeta del proyecto en la ruta del contenedor **Apache**, inserta los datos en la base de datos del contenedor **MariaDB** y lanza el contenedor **PHPMyAdmin** conectado al de MariaDB.

6 Conslusiones

Podemos concluir que después del desarrollo de nuestra aplicación, hemos cumplido con los requisitos especificados.

6.1 Mejora de la accesibilidad a la seguridad digital

Utilizando nuestra herramienta, podemos dar por supuesto que los usuarios crearán el contenido acorde a su rol profesional, esto mejorará con creces la comprensión de los artículos técnicos para la seguridad digital.

Por esto creemos que hemos conseguido afrontar y sobrepasar el primer problema más recurrente en el campo de la seguridad digital, *los artículos de seguridad digital son muy técnicos e incomprensibles.*

Gracias a nuestra herramienta, los usuarios, especificados por su rol, tendrán la información más clara y debido a la clasificación por niveles, se encontrará con artículos con nivel que le ayudará a saber la dificultad del mismo, lo que hará que el usuario tenga más facilidad de comprensión sobre el tema, eligiendo artículos que sea capaz de comprender.[8]

6.2 Ofrecer contribuciones como usuario

Las funcionalidades de uso de la aplicación como usuario, son suficientes para poder utilizar la plataforma, desde el punto en que estamos, aun que la aplicación tiene posibilidad de desarrollo, el usuario tiene todas las funcionalidades necesarias implementadas. Podrá consultar los artículos creados por la comunidad, hacer exámenes para entrenar y aprender, subir de nivel para escalar en los niveles de rol para poder crear contenido de mayor dificultad.

También quedan terminadas las funcionalidades de votaciones de los artículos creados para su posterior publicación haciendo crecer la comunidad.

Las funcionalidades de creación y modificación de los datos de la cuenta de usuario también están accesibles desde la vista de perfil15.

6.3 Ofrecer colaboración como desarrollador

Por otro lado, completamos las funcionalidades necesarias para que el desarrollador pueda contribuir y desarrollar con nosotros.

[8]Dejando acceso a todos los artículos de todos los niveles pues nuestra comunidad debe gozar de la mayor libertad posible.

Para ello, tenemos el despliegue en docker del entorno de desarrollo de la manera más simple y con bajo consumo posible. Con esto, el desarrollador podrá rápidamente ponerse manos a la obra en un entorno seguro y cuidado, con una estructura realista del entorno en el que se desplegará la herramienta en público.

Tendrá a su disposición la posibilidad de contribuir en **GitHub** haciendo un fork de nuestro repositorio y un pull request a la hora de querer mergear sin conflictos y asegurando que cualquier usuario desde cualquier parte pueda sugerir sus modificaciones a los administradores del proyecto.

6.4 Garantizar mejoras de seguridad

Para terminar de desarrollar un proyecto sólido y seguro, he aportado las siguientes medidas de seguridad para resolver ciertas vulnerabilidades.

Tras la resolución de las vulnerabilidades más importantes, quedan resueltas **114** de las **227** vulnerabilidades. Lo que supone una mejora de más del **50%** de las vulnerabilidades y siendo esta mitad, la mitad de más peso.

En lo sucesivo seguiré resolviendo vulnerabilidades para asegurar aún más la aplicación.

7 Bibliografía

A continuación detallo la bibliografía diferenciando herramientas utilizadas, documentación y aquellas fuentes utilizadas para la investigación.

7.1 Herramientas

En esta sección especifico las fuentes utilizadas como herramientas para el desarrollo del proyecto.

- DBdesigner: esta web ha sido utilizada para diseñar el esquema que se presenta en la figura 5.
- GenerateData: esta web se ha utilizado para generar los datos utilizados en la población masiva de la aplicación.
- Github: esta web se ha utilizado para llevar un control de las versiones del proyecto durante su desarrollo.
- draw.io: he utilizado esta web para generar el diagrama de contenedores docker.
- LaTeX: he escrito esta memoria usando LaTeX y el entorno gráfico TeXstudio.
- OWASP ZAP: herramienta de auditoría de seguridad para aplicaciones web.
- Kali Linux: distribución de Linux basada en Debian con herramientas orientadas a la auditoría informática de seguridad.
- Docker: software de virtualización de contenedores.

7.2 Documentación e investigación

En esta sección detallaré las fuentes utilizadas para la investigación del proyecto.

- Guía de autodefensa digital de SoftCatala: https://autodefensa.softcatala.cat
- Security in a box: https://securityinabox.org
- Hacklab de Ingobernable: https://hacklab.ingobernable.net
- Security Education Companion EFF: https://sec.eff.org
- AWS Docker containers AWS: https://aws.amazon.com/es/getting-started/tutorials/deploy-docker-containers/
- Espacio de Ciberseguridad y programación segura en sitios web: guía de seguridad web.

8 English version

To meet the objectives demanded by the University, I will translate the **introduction** and the **conclusions**, a translated **abstract** could be found at the beginning of this document.

8.1 Introduction

In this section we will see the proyect motivation, the goals to achieve and similar existing tools.

8.1.1 Objectives

To solve the mentioned problems, we should plan the objectives detailed below.

Improve the online security accesibility As mentioned before, thenks to the internet. we are able to comunicate and inform us about many different topics, but nevertheless those topics are too technical so **Kapi** should be a tool where all the information will be writen by and to a professional of an specified professional sector. We will be focused on three professional sectors (**lawyers**, **journalists** and **computer engineers**). This way we will try to break with the understanding problem, in this case, they will be able to find and understand what they need because the found articles will be written by collegues without being techcnical or in the oposite, too short and hard to understand.

To improve this, we want to create an enviroment where each article is written by a specific professional profile, and just by users with this professional profile. Before being validated, it will pass through a validation phase by the community. It will be subject to revisions and modifications until get its final shape to be published.

In any case, both an article in the revision phase ant the published one, will be able to be modified by other user with the same role.

User contribution The tool will offer the necessary ways for any user to obtain information about the field they want, but can only create articles of the professional role to which the user belongs. We will also provide a system of levels to classify users from exams so that they can only create/-modify content at their highest level.

Thus, users will not be able to modify articles from other users with a higher level and will have the opportunity to acquire that level by carrying

out examination tests.

The modifications and creations, before being published, are submitted to a voting phase, where the users decide which articles and modifications of articles are published.

Therefore, the contributions as a user will be:

- Create articles of maximum your level.
- Modify articles of your role, of a lower or equal level to that of the user.
- Voting to decide which articles are published and which are not.

Contributions as a developer **Kapi**, also offers the facility for internal project developers to work comfortably in a team, as well as the opportunity for anyone who wants to collaborate in a free software community, an easy to understand way to download the source code, launch it in a realistic environment to do the relevant tests before accepting the collaborator's code in the main project.

For this the tool will have public access to a repository based on a version controller and a way to virtualize the service in a simple and understandable way.

In short, the developer will have:

- Public repository with version control.
- Project virtualization method for test environment.

With a comfortable and light virtualization, I will make the development more homogeneous and portable. This way the development will be more comfortable and stable.

In addition, I will look for the lightest and most economical way to facilitate the realistic deployments of the platform.

Contributions to the improvement of application security I will analyze the tool for security vulnerabilities with web security auditing tools, looking for vulnerabilities in order to solve them and ensure the highest protection to users and a method of resolving vulnerabilities to the development community.

8.1.2 Work plan

The main problem to be solved by this project, is to improve the accessibility to information about online security that we find on the Internet.

Internet has a great amount of very good and very useful information, but sometimes we find that this information is explained in a very technical way and sometimes complex to understand even for people accustomed to the use of new technologies.

For this, **Kapi** is a website in which we differentiate users by professional roles, so that each user will find information written by people in their profession, which provides a more colloquial language on the technicalities, more personalized explanations (the knowledge in the gremios is usually similar, so the one who writes the article, will explain the more complex technicalities to the readers of his gremio in a way that is more understandable).

Kapi is a tool oriented as much as possible to the free software community. To achieve this, we will develop the whole tool with free software, with images and content with free license and everything will be public in a **git** repository, in our case hosted in GitHub.

This repository should be prepared to work comfortably with it and that the contributor is able to test his changes in his system in a realistic way.

The following planning remains:

1. Improve accessibility to online security information.
2. Differentiate users by professional roles, providing a more comfortable communication and adapted to the guild.
3. Implement a development methodology for the free software community.
4. Implement a portable and uniform deployment to develop over a real architecture.

8.1.3 Working Methodology

This web application has been developed by two students from the Faculty of Computer Science of the Complutense University of Madrid, **Álvaro Asenjo Torrico** and **Javier Martín Villarreal**.

Due to academic reasons, we plan to develop this project with a perspective to present it in consecutive years. From the beginning we worked

together separating the tasks in a clear way because my colleague would present a year before me and we wanted the tasks to be well differentiated for this reason.

This is why my partner would be in charge of developing the **front-end**.

On the other hand, I would be in charge of the tasks of **architectural implementation** of the project, as well as the **structure of the directory tree** that we will implement for the development, the **back-end** of the application (those areas of code that run on the server to return the data to the client), the definition of the **database**, the test **data population** on the database (to get an idea of the result of the web with content), the **virtualization** of this architecture for its portability and the study of the **security** of both the virtualized system and the application.

I will only mention the work developed by my colleague and I will try to detail only the tasks that I have developed on this tool.

Between the two of us, we decided on the flow we were going to develop to be able to access the articles on the web, be able to vote for the articles proposed by the community, be able to carry out examinations on the platform to ascend through the levels and the management of user data, both modification and visualisation of the same.

Next I explain in detail the development of the tasks developed by my collegue.

Front-end contribution My colleague, was responsible for developing the front of the application, all those parts that will run on the client.

To do this, he decided to implement it with PHP, HTML, JavaScript, CSS and Bootstrap to stylize the page.

He developed all the views of the application and we decided on the variables and methods we would use for the **communication** between the front-end and the back-end, uniformly communicating the client with the server.

8.2 Conslusions

We can conclude that after the develop of our application, we achieve our goals, acording to the specified requirements.

8.2.1 Improved the digital security accesibility

Using our tool, we can asume that the users will create content acording to their profession skills, this will improve so much the tecnic understanding of the articles about digital security.

This is why we consider thet we earned the main requirement, the most usual in digital security, *digital security articles, are complex and tecnical.*

Thanks to our tool, users, specified by their own role, will have better information due to the level clasification, their will find articles with levels to know the dificulty of it, this will make the user improve his understanding skills, choosing articles acording to their levels.[9]

8.2.2 Offer user contributions

The user usage of the application functionalities, are enough to use our platform, even the application has more develop posibilities, all the needed user functionalities are already implemented. Their could read the community articles, their could complete exams to train and learn, level up and reach the higher levels to create harder content.

Functionalities like voting articles already created to its future publication, are also implemented.

Also the ones of creation and modification od the account user data, accesible from the profile view.15.

8.2.3 Offer development contributions

In the other hand, we have complete all needed functionalities for the developer to contribute and develope with our community.

To achieve this goal, I have made the deploiment of a docker develompent enviroment in the easiest way and with the highest performance. With this, the developer will be able to get to work quickly in a secured and care enviroment with a realistic enviroment structure like the public deploiment one.

They will have the choice to colaborate on **GitHub** forking our repository and making a pull request to merge with no conflicts and leaving to any user sugest their modifications to the proyect admin.

[9]Leaving access to all the articles of all levels to give freedom to our community.

8.2.4 Guarantee security improvements

To make a solid and secured finish of the proyect, I have contributed with the following security measures to resolve certain vulnerabilities.

After the resolution of the most important vulnerabilities, **114** of **227** vulnerabilities are solved. What means the improvements of more than the **50%** what means more than a half, and the most important ones.

In the future I will follow solving vulnerabilities to improve the application security.

9 Apéndice

9.1 Capturas de pantalla de la aplicación

Estas capturas, son el resultado de las queries (desarrolladas por mi) que devuelve el back-end, renderizadas en el front-end web ()desarrollado por mi compañero).

Figura 8: Signin

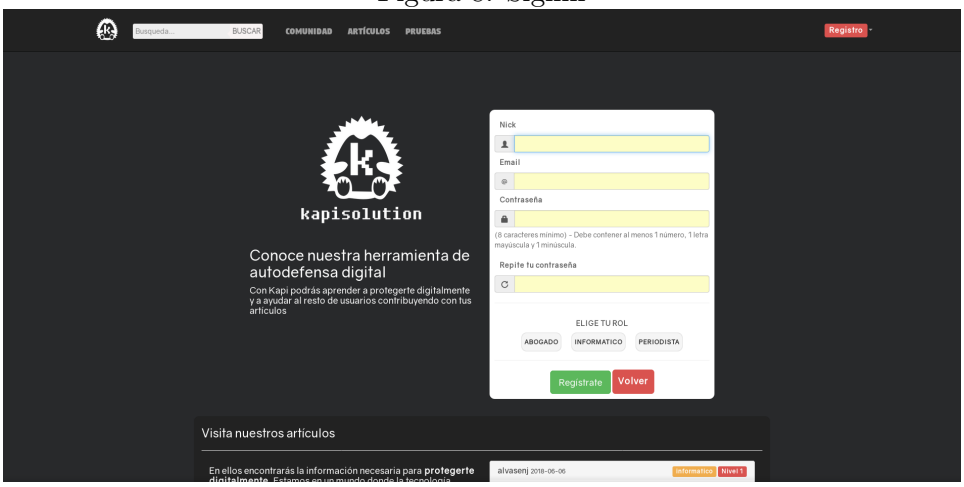


Figura 9: Login

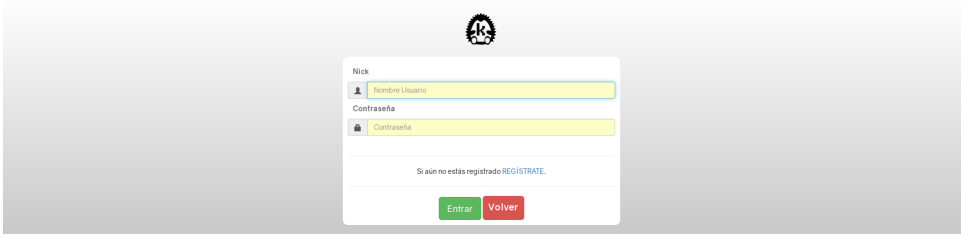


Figura 10: Artículos

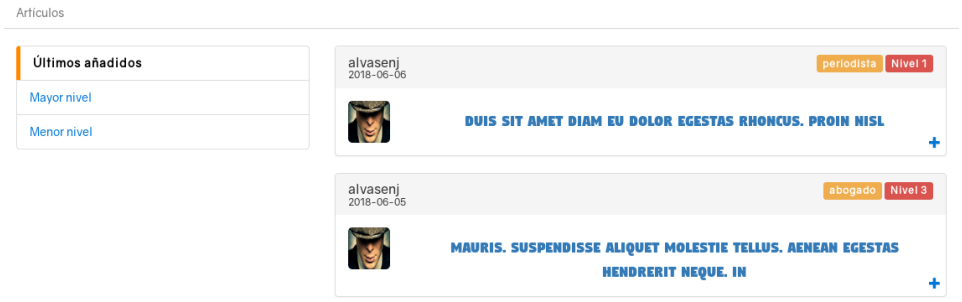


Figura 11: Modificaciones

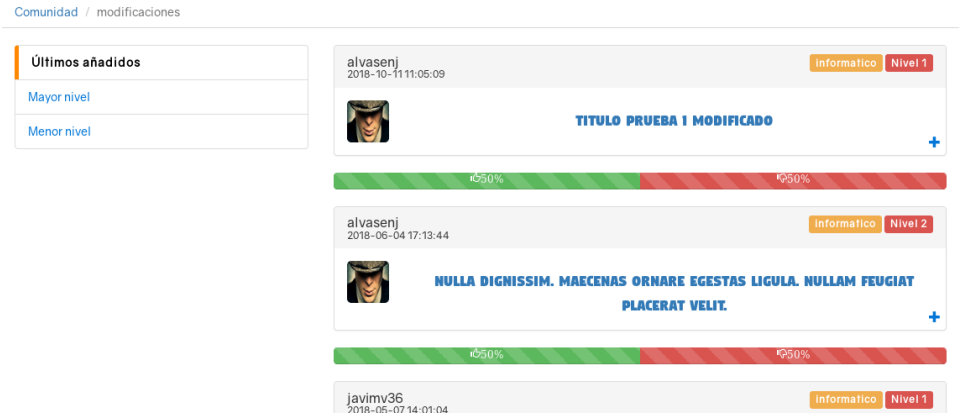


Figura 12: Creaciones

Figura 13: Crear artículo

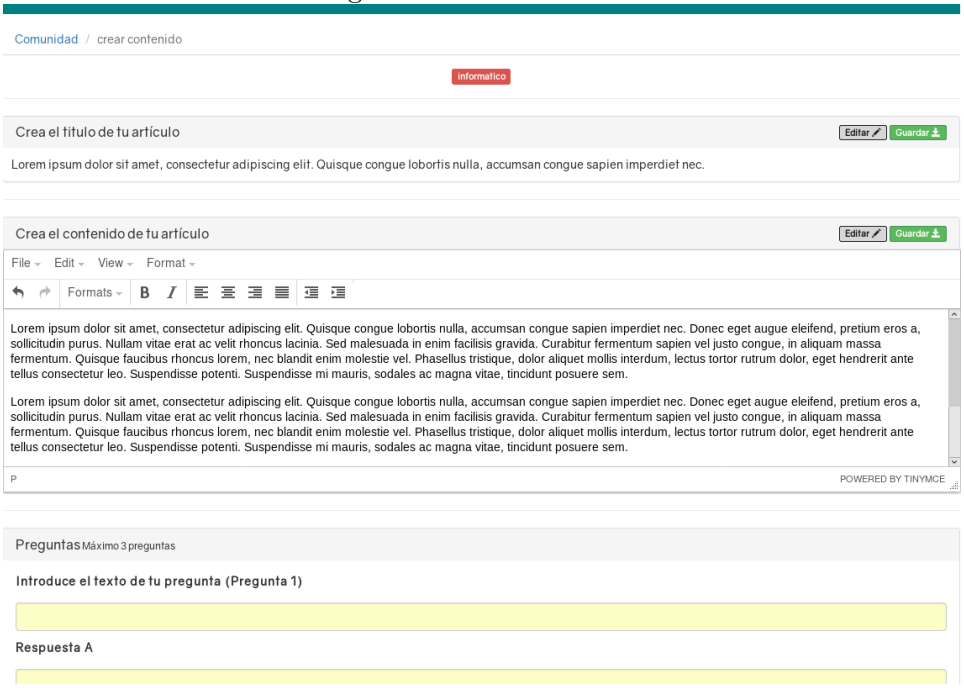


Figura 14: Modificación

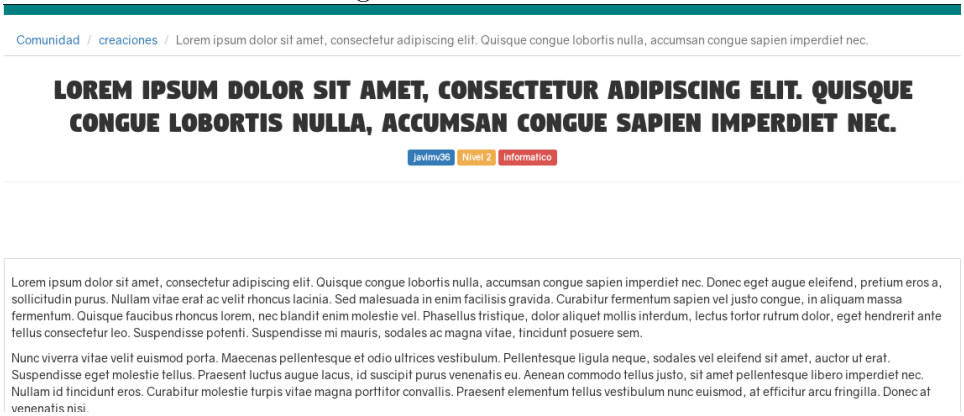


Figura 15: Perfil

Figura 16: Exámenes

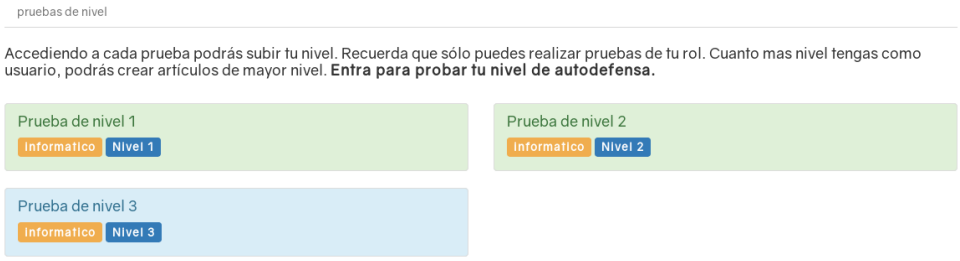


Figura 17: Examen

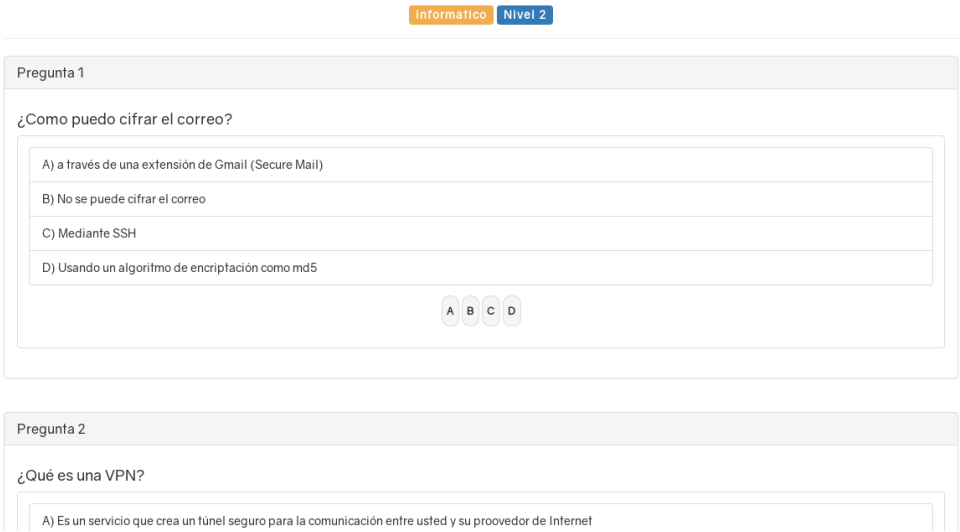


Figura 18: Alertas detectadas con OWASP ZAP